

AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application.

1 1. (Cancelled)

1 2. (Currently Amended) The method of claim [[1]]11, wherein the computer further
2 includes a CPU, wherein the virtual machine monitor is in control of the CPU prior to the
3 runtime virtualization of the I/O device.

1 3. (Currently Amended) The method of claim [[1]]11, wherein the virtualization is
2 performed transparently to [[the]]an operating system.

1 4. (Currently Amended) The method of claim [[1]]11, wherein the I/O device is
2 compatible with the virtualized I/O device.

1 5. (Cancelled)

1 6. (Currently Amended) The method of claim [[5]]11, further comprising
2 configuring [[the]] hardware to trap I/O accesses, and enabling the virtual machine monitor to
3 emulate the I/O device in response to the traptrapped I/O accesses.

1 7. (Original) The method of claim 6, wherein the virtual machine monitor uses
2 memory management to trap the I/O accesses.

1 8. (Currently Amended) The method of claim [[5]]15, wherein the virtual machine
2 monitor can commencecommences the emulationvirtualization between I/O sequences.

1 9. (Currently Amended) The method of claim 8, wherein the virtual machine
2 monitor commences emulation the virtualization by intercepting I/O accesses; wherein the
3 virtual machine monitor uses the intercepted I/O accesses to update a state machine, whereby the
4 state machine reflects a state of the I/O device; and wherein the virtual machine monitor
5 examines transitions in the state machine to determine whether the I/O device is in the middle of
6 an I/O sequence.

1 10. (Currently Amended) The method of claim [[5]]15, wherein further comprising
2 the virtual machine monitor can commence the emulation in the middle of commencing the
3 virtualization during an I/O sequence.

1 11. (Currently Amended) The method of claim 5In a computer including an I/O
2 device, a method comprising using a virtual machine monitor to commence virtualization of the
3 I/O device at runtime, wherein runtime is a period of execution in the computer after boot and
4 before shutdown of the computer, wherein the virtual machine monitor uses a state machine to
5 determine determines whether the I/O device is in the middle of performing an I/O sequence, and
6 delays commencing emulation the virtualization until the state machine indicates virtual machine
7 monitor determines that the I/O sequence has completed.

1 12. (Currently Amended) The method of claim [[1]]11, wherein the runtime
2 virtualization includes using the virtual machine monitor to emulate I/O device interrupts.

1 13. (Currently Amended) The method of claim [[1]]11, wherein I/O device interrupts
2 are directed to an operating system prior to the runtime virtualization of the I/O device; and
3 wherein the I/O device interrupts are directed to the virtual machine monitor during and after the
4 virtualization of the I/O device.

1 14. (Currently Amended) The method of claim [[1]]11, wherein the virtual machine
2 monitor temporarily pauses an I/O sequence by emulating the I/O device as being busy.

1 15. (Currently Amended) The method of claim 11 in a computer including an I/O
2 device, a method comprising:

3 using a virtual machine monitor to commence virtualization of the I/O device at runtime,
4 wherein runtime is a period of execution in the computer after boot and before shutdown of the
5 computer, wherein the I/O device has multiple modes of operations;

6 wherein the virtual machine monitor determines determining the mode of the I/O device
7 prior to commencing the virtualization; and

8 wherein the virtual machine monitor restores restoring the determined mode of [[the]]
9 operation after the virtualization.

1 16. (Currently Amended) The method of claim [[1]]11, further comprising
2 devirtualizing the I/O device at runtime following the runtime virtualization.

1 17. (Currently Amended) In a computer including hardware, a method comprising:
2 running a virtual machine monitor running on the hardware[[],];

3 running an operating system running on the virtual machine monitor,
4 wherein the hardware including includes an I/O device, and the I/O device is already

5 virtualized by the virtual machine monitor[[],]; and
6 a method comprising devirtualizing the I/O device at runtime, wherein runtime is a period
7 of execution in the computer after boot and before shutdown of the computer.

1 18. (Original) The method of claim 17, wherein the devirtualization is performed
2 transparently to the operating system.

1 19. (Original) The method of claim 17, wherein the devirtualization includes stopping
2 I/O device emulation at runtime.

1 20. (Original) The method of claim 17, wherein the virtual machine monitor emulates
2 the I/O device prior to devirtualization; and wherein the devirtualization includes allowing the

3 virtual machine monitor to temporarily stop the operating system from commencing a new I/O
4 sequence.

1 21. (Original) The method of claim 20, wherein the virtual machine monitor
2 temporarily stops the operating system by emulating the I/O device as being in a "busy" or
3 "device not ready" state.

1 22. (Original) The method of claim 20, wherein the virtual machine monitor bounds
2 the amount of time the operating system processing is temporarily stopped.

1 23. (Currently Amended) The method of claim 20, further comprising:
2 ~~wherein~~ the virtual machine monitor ~~[[logs]]logging~~ I/O accesses by the operating system
3 to the I/O device during devirtualization, and
4 ~~replays-replaying~~ the log to the I/O device after devirtualization, ~~[[whereby]]wherein~~ the
5 I/O accesses by the operating system are deferred during the devirtualization of the I/O device.

1 24. (Original) The method of claim 17, wherein the virtual machine monitor waits for
2 I/Os initiated by the virtual machine monitor's driver for the I/O device to complete, and for all
3 expected interrupts from the device to arrive, before ceasing device emulation.

1 25. (Currently Amended) The method of claim 17, further comprising re-directing
2 interrupts from interrupt handlers in the virtual machine monitor to interrupt handlers in the
3 operating system after performing the devirtualizing.

1 26. (Currently Amended) The method of claim 17, further comprising, after
2 performing the devirtualizing, configuring the hardware so ~~[[the]]~~ accesses by the operating
3 system to the I/O device no longer trap to the virtual machine monitor.

1 27. (Currently Amended) The method of claim 17, wherein the I/O device has
2 multiple modes of operations[;], the method further comprising:
3 wherein the virtual machine monitor determines determining the mode of the I/O device
4 prior to commencing the devirtualization; and
5 wherein the virtual machine monitor restores restoring the determined mode of [[the]]
6 operation after devirtualization.

1 28. (Currently Amended) The method of claim 17, further comprising
2 virtualizing wherein the I/O device is virtualized at runtime again after [[having]] performing the
3 devirtualizing been devirtualized at runtime.

1 29. (Cancelled)

1 30. (Currently Amended) The computer of claim [[29]]35, wherein the I/O device is
2 compatible with the virtualized I/O device.

1 31. (Cancelled)

1 32. (Currently Amended) The computer of claim [[31]]35, further comprising
2 configuring wherein the hardware is configured to trap I/O accesses, and enabling the virtual
3 machine monitor is enabled to emulate the I/O device in response to the trapped I/O
4 accesses traps.

1 33. (Currently Amended) The computer of claim 32, wherein the virtual machine
2 monitor [[uses]]is configured to use memory management to trap the I/O accesses.

1 34. (Cancelled)

1 35. (Currently Amended) The computer of claim 34A computer comprising:
2 hardware including an I/O device; and
3 computer memory encoded with a virtual machine monitor for running on the hardware
4 and commencing virtualization of the I/O device at runtime, wherein runtime is a period of
5 execution in the computer after boot and before shutdown of the computer,

6 wherein the virtual machine monitor [[uses]]is configured a-state-machine-to determine

7 whether the I/O device is in the middle-of-performing an I/O sequence, and delays-to delay

8 commencing emulationthe virtualization until the state machine indicates virtual machine

9 monitor determines that the I/O sequence has completed.

1 36. (Currently Amended) The computer of claim [[31]]35, wherein the virtual
2 machine monitor is configured to temporarily pause the I/O sequence by emulating the
3 I/O device as being busy.

1 37. (Currently Amended) The computer of claim [[29]]35, wherein the runtime
2 virtualization includes using the virtual machine monitor to emulate I/O device interrupts.

1 38. (Currently Amended) A computer comprising:
2 hardware including an I/O device; and
3 computer memory encoded with a virtual machine monitor for devirtualizing the I/O
4 device at runtime, wherein runtime is a period of execution in the computer after boot and before
5 shutdown of the computer.

1 39. (Currently Amended) The computer of claim 38, wherein the virtual machine
2 monitor is configured to emulate emulates the I/O device prior to commencing the
3 devirtualization; and wherein the virtual machine is configured to commence commences the
4 devirtualization by temporarily stopping an operating system running on the virtual machine
5 monitor from commencing a new I/O sequence.

1 40. (Currently Amended) The computer of claim 39, wherein the virtual machine
2 monitor is configured to temporarily stop the operating system by emulating the I/O device
3 as being in a "busy" or "device not ready" state.

1 41. (Currently Amended) The computer of claim 39, wherein the virtual machine
2 monitor is configured to bound bounds the amount of time the operating system processing is
3 temporarily stopped.

1 42. (Currently Amended) The computer of claim [[39]]38, wherein the virtual
2 machine monitor [[logs]]is configured to log I/O accesses by an operating system to the I/O
3 device during devirtualization, and to replay replays the log to the I/O device after
4 devirtualization.

1 43. (Currently Amended) The computer of claim 39, wherein the virtual machine
2 monitor is configured to wait waits for I/Os initiated by a virtual machine monitor driver for the
3 I/O device to complete, and for all expected interrupts from the I/O device to arrive, before
4 ceasing device emulation.

1 44. (Currently Amended) The computer of claim 38, further comprising configuring
2 wherein the hardware is configured so operating system accesses to the I/O device no longer trap
3 to the virtual machine monitor after the devirtualization.

1 45. (Currently Amended) The computer of claim 38, wherein the I/O device has
2 multiple modes of operations; wherein the virtual machine monitor is configured to determine
3 determines the mode of the I/O device prior to commencing the devirtualization; and wherein the
4 virtual machine monitor is configured to restore restores the determined mode of [[the]]
5 operation after the I/O device has been devirtualized.

1 46. (Currently Amended) The computer of claim 38, wherein the virtual machine
2 monitor [[can]] is configured to further virtualize the I/O device after having devirtualized the I/O
3 device at runtime.

1 47. (Cancelled)

1 48. (Currently Amended) The article of claim [[47]]52, wherein the virtualization
2 includes commencing I/O device emulation at runtime.

1 49. (Currently Amended) The article of claim 48, wherein the software includes a
2 virtual machine monitor; and wherein the software configures the hardware to trap I/O accesses,
3 and enables the virtual machine monitor to emulate the I/O device in response to the trapped I/O
4 devices traps.

1 50. (Previously Presented) The article of claim 49, wherein the virtual machine
2 monitor uses memory management to trap the I/O accesses.

1 51. (Cancelled)

1 52. (Currently Amended) The article of claim 51An article for a computer including
2 an I/O device, the article comprising computer-readable memory encoded with a virtual machine
3 monitor for causing the computer to commence virtualization of the I/O device at runtime,
4 wherein runtime is a period of execution in the computer after boot and before shutdown of the
5 computer, wherein the virtual machine monitor includes a state machine for determining
6 determines whether the I/O device is in-the-middle-of performing an I/O sequence, the virtual
7 machine monitor delaying the commencement of the emulation virtualization until the state
8 machine indicates virtual machine monitor determines that the I/O sequence has completed.

1 53. (Original) The article of claim 52, wherein the virtual machine monitor
2 temporarily pauses the I/O sequence by emulating the I/O device as being busy.

1 54. (Currently Amended) The article of claim [[47]]52, wherein the software includes
2 a virtual machine monitor for emulatingemulates I/O device interrupts during the runtime
3 virtualization.

1 55. (Cancelled)

1 56. (Currently Amended) An article for a computer including an I/O device, the
2 article comprising computer-readable memory encoded with softwarea virtual machine monitor
3 for causing the computer to devirtualize the I/O device at runtime, wherein runtime is a period of
4 execution in the computer after boot and before shutdown of the computer.

1 57. (Original) The article of claim 56, wherein the devirtualization includes ceasing
2 emulation of the I/O device at runtime.

1 58. (Currently Amended) The article of claim 57, wherein the software includes a
2 virtual machine monitor; and wherein the devirtualization includes temporarily stopping an
3 operating system running on the virtual machine monitor from commencing a new I/O sequence.

1 59. (Original) The article of claim 58, wherein the virtual machine monitor
2 temporarily stops the operating system by emulating the I/O device as being in a "busy" or
3 "device not ready" state.

1 60. (Original) The article of claim 58, wherein the virtual machine monitor bounds
2 the amount of time the operating system processing is temporarily stopped.

1 61. (Currently Amended) The article of claim 57, wherein ~~the software includes a~~
2 ~~virtual machine monitor for ceasing the emulation;~~ the virtual machine monitor ~~waiting~~ waits for
3 I/Os initiated by a virtual machine monitor driver for the I/O device to complete, and for all
4 expected interrupts from the I/O device to arrive, before ceasing device emulation.

1 62. (Currently Amended) The article of claim 56, wherein the ~~software includes a~~
2 virtual machine monitor logs for logging I/O accesses by an operating system to the I/O device
3 during devirtualization, and ~~replaying~~ replays the log to the I/O device after devirtualization.

1 63. (Currently Amended) The article of claim 56, wherein the ~~software includes a~~
2 virtual machine monitor, ~~the software configuring~~ configures the hardware so operating system
3 accesses to the I/O device do not trap to the virtual machine monitor.

1 64. (Currently Amended) The article of claim 56, wherein the I/O device has multiple
2 modes of operations; and wherein the ~~software includes a~~ virtual machine monitor ~~for~~
3 ~~determining~~ determines the mode of the I/O device prior to commencing devirtualization; and
4 ~~restoring~~ restore the determined mode of [[the]] operation after the I/O device has been
5 devirtualized.

1 65. (Cancelled)

1 66. (Currently Amended) The article of claim [[65]]56, wherein the virtual machine
2 monitor [[can]]causes the computer to further virtualize the I/O device after having devirtualized
3 the I/O device at runtime.